

```

1.
2. 64K DATA & OP CODE CONVERTER AB or BB after substitution
3.      ↵=right arrow
4. JIM DUNSON s=leave 1 space
5. 13900 RIVER ROAD x=multiply symbol
6. PERDIDO KEY FL 32507 d=divide symbol
7. (904) 492-1470 ÷=more than symbol
8.      /<=less than symbol
9.
10 CLEAR :NT=0;BC=127;FC=8;PRINT "DATA & OP CODE
    CONVERTER";BOX -2,25,140,I,1
120 PRINT :PRINT "ENTER MINUS SIGN FIRST WITH NEGATIVE
    DECIMAL."
130 PRINT :PRINT "ALL ENTRIES: ENTER ZEROS LEADING, IF NEEDED,
    TO TOTAL REQUIRED DIGITS."
140 CY=-39;PRINT ".....PUSH ANY KEY.....";IF KPCLEAR
150 PRINT :PRINT "SELECT KNOWN RADIX";PRINT :PRINT "ss(1) BINARY"
    :PRINT "ss(2) DECIMAL";PRINT "ss(3) HEXADECIMAL"
160 B=KP;IF (B=49)+(B=50)+(B=51)GOTO 80
170 GOTO 230
180 CLEAR :IF B=49A=16;GOSUB 310;PRINT "BINARY";GOTO 150
190 IF B=51A=4;GOSUB 310;PRINT "HEXADECIMALS";GOTO 190
200 .ENTER DECIMAL
210 A=5;C=0;GOSUB 310;PRINT "DECIMAL";CX=49;FOR F=20022TO 20030STEP
    2;GOSUB 250;IF K=-3CX=49;PRINT "s";CX=43;PRINT "-";C=2;F=2;NEXT F
120 NEXT F:D=(Kx10000)+(Lx1000)+(Mx100)+(Nx10)+O;IF C=2D=-D
130 PRINT :GOSUB 330;GOSUB 370;GOTO 270
140 .ENTER BINARY
150 FOR F=20022TO 20052STEP 2;GOSUB 250;IF (%F)÷1)+(F)÷0GOTO 230
160 IF (F=20028)+(F=20036)+(F=20044)TV=32
170 NEXT F;PRINT :GOSUB 370;GOSUB 430;GOTO 270
180 .ENTER HEX
190 CX=55;FOR F=20014TO 20020STEP 2;A=KP;IF A÷64IF A÷71(F)=A-55;TV=A;
    NEXT F
200 IF A÷47IF A÷58(F)=A-48;TV=A;NEXT F
210 IF F÷20020PRINT :GOSUB 430;GOSUB 330;GOTO 270
220 .WRONG KEY
230 CLEAR :CY=0;PRINT "INVALID ENTRY";FOR F=1TO 100;
    NEXT F;CLEAR :GOTO 50
240 .TV B. & D. ENTRY
250 (F)=KP-48;TV=-(F)+48;RETURN
260 .DISPLAY & OPTIONS
270 CY=34;PRINT "DATA & OP CODE EQUIVELANTS";BOX -1,29,159,1,I;CY=-39
280 BOX -1,-32,159,1,I;PRINT "(I) NEW RADIX(s) REPEAT";IF KP=49CLEAR
    GOTO 50
290 GOTO 80
300 .PRE-ENTRY
310 PRINT :PRINT "#1,"sENTERS",A,"sDIGITS,";PRINT :RETURN
320 .DECIMAL TO BINARY
330 E=D;A=0;IF E÷0 E=E÷32767+I;A=1
340 FOR F=20052TO 20022STEP -2;E=E÷2;%(F)=RM;NEXT F;IF A=1K=1
350 FRNT1 :PRINT #1,"BINARYs",K,L,M,N,"s",O,P,Q,R,"s",S,T,U,V,"s",
    W,X,Y,Z;RETURN
360 .BINARY TO HEX
370 J=Z+(2xY)+(4xX)+(8xW);I=V+(2xU)+(4xT)+(8xS)
380 H=R+(2xQ)+(4xP)+(8xO);G=N+(2xM)+(4xL)+(8xK)
390 .TV HEX. HOLD GHIJ
400 PRINT :PRINT "HEXADECIMAL";CX=55;FOR F=20014TO 20020STEP 2;
    A=-(F)÷48;IF A÷57A=A+7
410 TV=A;NEXT F;PRINT :RETURN
420 .HEX TO DECIMAL
430 A=D;IF G÷7G=G-8;A=1
440 D=(4096xG)+(256xH)+(16xI)+J;IF A=1D=D-32767-1
450 PRINT :PRINT "DECIMAL",#19,D;RETURN

```

TO RUN IN BB SUBSTITUTE AS FOLLOWS:

```

110 FOR F=20022TO 20030 substitute FOR F=20098TO 20106
150 FOR F=20022TO 20052 substitute FOR F=20098TO 20128
160 IF (F=20028)+(F=20036)+(F=20044) sub IF (F=20104)+(F=20112)+(F=20120)
190 FOR F=20014TO 20020 substitute FOR F=20090TO 20096
210 IF F÷20020 substitute IF F÷20098
340 FOR F=20052TO 20022 substitute FOR F=20128TO 20098
400 FOR F=20014TO 20020 substitute FOR F=20090TO 20096

```

ON:16 APR 1984

```

1 .
2 . 64K DATA & OP CODE
3 . CONVERTER
4 .
5 . JIM DUNSON
6 . 1#900 RIVER ROAD
7 . PERDIDO KEY19526 32507
8 . (904) 492-1470
9 .
10 CLEAR ;NT=0;BC=127;FC=8;PRINT ;PRINT ">DATA & OP CODE CONVERTER<";BOX -2,25,1
11 40,1,1
12 20 PRINT ;PRINT "a ENTER MINUS SIGN FIRST WITH NEGATIVE DECIMAL.
13 30 PRINT ;PRINT "a ALL ENTRIES: ENTER ZEROS LEADING, IF NEEDED, TO TOTAL REQ
14 UIRED DIGITS.
15 40 CY=-39;PRINT ".....PUSH ANY KEY.....";IF KPCLEAR
16 50 PRINT ;PRINT "a SELECT KNOWN RADIX";PRINT ;PRINT " (1) BINARY";PRINT " (2)
17 DECIMAL";PRINT " (3) HEXADECIMAL
18 60 B=KP;IF (B=49)+(B=50)+(B=51)GOTO 80
19 70 GOTO 230
20 80 CLEAR ;IF B=49A=16;GOSUB 310;PRINT "BINARY ";GOTO 140
21 90 IF B=51A=4;GOSUB 310;PRINT "HEXADECIMAL ";GOTO 180
22 100 .ENTER DECIMAL
23 110 A=5;C=0;GOSUB 310;PRINT "DECIMAL";CX=49;FOR F=20022TO 20030STEP 2;GOSUB 250
24 ;IF K=-3CX=49;PRINT " ";CX=43;PRINT "-";C=2;F=F-2;NEXT F
25 120 NEXT F;D=(Kb10000)+(Lb1000)+(Mb100)+(Nb10)+0;IF C=2D=-D
26 130 PRINT ;GOSUB 330;GOSUB 370;GOTO 270
27 140 .ENTER BINARY
28 150 FOR F=20022TO 20052STEP 2;GOSUB 250;IF (%(F)>1)+(%(F)<0)GOTO 230
29 160 IF (F=20028)+(F=20036)+(F=20044)TV=32
30 170 NEXT F;PRINT ;GOSUB 370;GOSUB 430;GOTO 270
31 180 .ENTER HEX
32 190 CX=55;FOR F=20014TO 20020STEP 2;A=KP;IF A>84IF A<71%(F)=A-55;TV=A;NEXT F
33 200 IF A>47IF A<58%(F)=A-48;TV=A;NEXT F
34 210 IF F>20020PRINT ;GOSUB 430;GOSUB 330;GOTO 270
35 220 .WRONG KEY
36 230 CLEAR ;CY=0;PRINT ">>>> INVALID ENTRY <<<<";FOR F=1TO 100;NEXT F;CLEAR ;G
37 240 0TO 50
38 240 .TV B. & D. ENTRY
39 250 %(F)=KP-48;TV=%(F)+48;RETURN
40 260 .DISPLAY & OPTIONS
41 270 CY=34;PRINT "DATA & OP CODE EQUIVELANTS";BOX -1,29,159,1,1;CY=-39
42 280 BOX -1,-32,159,1,1;PRINT "(1) NEW RADIX ( ) REPEAT";IF KP=49CLEAR ;GOTO 50
43
44 290 GOTO 80
45 300 .PRE-ENTRY
46 310 PRINT ;PRINT #1,"a ENTER ",A," DIGITS,";PRINT ;RETURN
47 330 .DECIMAL TO BINARY
48 340 E=D;A=0;IF E<0E=E+32767+1;A=1
49 350 FOR F=20052TO 20022STEP -2;E=E<2;%(F)=RM;NEXT F;IF A=1K=1
50 360 PRINT ;PRINT #1,"BINARY ",K,L,M,N," ",O,P,Q,R," ",S,T,U,V," ",W,X,Y,Z;RETURN
51
52 370 .BINARY TO HEX
53 380 J=Z+(2bY)+(4bX)+(8bW);I=U+(2bV)+(4bT)+(8bS)
54 390 H=R+(2bQ)+(4bP)+(8bO);G=N+(2bM)+(4bL)+(8bK)
55 400 .TV HEX. HOLD GHJJ
56 410 PRINT ;PRINT "HEXADECIMAL";CX=55;FOR F=20014TO 20020STEP 2;A=-(F)+48;IF A>5
57 7A=A+7
58 420 TV=A;NEXT F;PRINT ;RETURN

```

```
430 .HEX TO DECIMAL
440 A=D; IF G>7G=G-8; A=1
450 D=(4096bG)+(256bH)+(16bI)+J; IF A=1D=D-32767-1
460 PRINT ;PRINT "DECIMAL", #19, D; RETURN
```

1. ?

2. Straight line

3. Math Teacher

4. Match My Number

3:

4:

$$A = 24576$$

$$B = -22858$$

$$-22773$$

$$\text{for } n = A + B$$

$$\% (n) = KP$$

$$n \leq N$$

$$\% (20050) = 6$$

$$i \text{ set}$$

$$NT = 0$$

$$\text{Run}$$

$$D = 7$$
$$E = 7$$
[illegible]

```

32000 FOR A=D TO E; IF %(A)c256#13 NEXT A
32010 N=N+1; A=A+2; @(N)=%(A); NEXT A
32020 INPUT ".FL#*0, ".SP"G;:PRINT ;M=0;GOSUB F;FOR A=D TO E;TV=%(A);B=%(A)c256;IF
(RM=13)+(RM=-243)GOSUB F
32030 IF (RM=110)+(RM=111)GOSUB H
32040 NEXT A;PRINT ;STOP
32050 T=0;FOR B=A TO A+5;IF (%(B)c256-53)c6=0 T=Tb10+RM+5;A=A+1;NEXT B
32060 FOR B=0 TO N;IF @(B)=T PRINT #0,BbG+0,;RETURN
32070 NEXT B;RETURN
32080 PRINT #4,MbG+0,;TV=32;M=M+1;A=A+2;RETURN

```

2

Dunson
#1

```

1 .
2 .
3 .
4 .
5 :RETURN
6 .MATH TEACHER
7 .J DUNSON
8 .REV 2/81
9 E=420;F=430;G=450;H=330;I=390;J=400;K=410;N=360;O=370;Q=440
10 NT=0;GOSUB F;CLEAR ;&(9)=168;GOSUB Q;PRINT ;PRINT ;PRINT "*" aMATH TEACH
ER_ <*>;CY=-24;PRINT " + - b c PUSH GO! c b - +";GOSUB E
20 FOR D=1TO 2000;IF &(23)=1GOSUB F;GOTO 40
30 NEXT D;FC=BC-7;GOTO 20
40 CLEAR ;PRINT ;PRINT ;PRINT "*" HOW MANY WOULD YOU <* LIKE THIS TIME?"
;GOSUB E
50 CX=-6;INPUT "P;IF (P<1)+(P>100)PRINT "a FROM 1 TO 100 PLEASE! _";GOTO 50
60 GOSUB F;CLEAR ;PRINT ;PRINT "> SELECT SKILL LEVEL <*>;PRINT ;GOSUB G;PRI
NT "1. (1 TO 5)";GOSUB G;PRINT "2. (1 TO 10)";GOSUB G;PRINT "3. (1 TO 12)
70 GOSUB G;PRINT "4. (5 TO 12)";GOSUB G;PRINT "5. (9 TO 12)";GOSUB G;PRINT "6.
(12 TO 20)";GOSUB E;M=KP;GOSUB F
80 CLEAR ;PRINT ;PRINT "> MAKE SELECTION <*>;PRINT ;GOSUB G;PRINT "1. A
DD";GOSUB G;PRINT "2. SUBTRACT";GOSUB G;PRINT "3. MULTIPLY";GOSUB G
90 PRINT "4. DIVIDE";GOSUB G;PRINT "5. MIXED";GOSUB E;T=KP
100 R=0;W=0;L=P+1;CLEAR
110 CLEAR ;IF R+W>P-1GOSUB F;GOTO 460
120 IF M=49A=RND (5);B=RND (5)
130 IF M=50A=RND (10);B=RND (10)
140 IF M=51A=RND (12);B=RND (12)
150 IF M=52A=RND (8)+4;B=RND (8)+4
160 IF M=53A=RND (4)+8;B=RND (4)+8
170 IF M=54A=RND (8)+12;B=RND (8)+12
180 IF T=49GOTO 250
190 IF T=50GOTO 270
200 IF T=51GOTO 290
210 IF T=52GOTO 310
220 S=RND (4);IF S=2GOTO 270
230 IF S=3GOTO 290
240 IF S=4GOTO 310
250 GOSUB H;GOSUB N;PRINT #1,A,"+",B,"=";;INPUT "C;IF A+B=CGOTO I
260 GOSUB J;PRINT #2,A+B;GOSUB K;GOTO 250
270 GOSUB H;GOSUB N;PRINT #1,A+B,"-",B,"=";;INPUT "C;IF A=CGOTO I
280 GOSUB J;PRINT #2,A;GOSUB K;GOTO 270
290 GOSUB H;GOSUB O;PRINT #1,A,"b",B,"=";;INPUT "C;IF AbB=CGOTO I
300 GOSUB J;PRINT #2,AbB;GOSUB K;GOTO 290
310 GOSUB H;GOSUB O;PRINT #1,AbB,"c",B,"=";;INPUT "C;IF A=CGOTO I
320 GOSUB J;PRINT #2,A;GOSUB K;GOTO 310
330 L=L-1;CX=-6;CY=38;PRINT #1,L;GOSUB Q;BOX 0,0,84,40,1;BOX 0,0,72,32,2;X=-15;
CY=0;IF A>9X=X-3
340 IF B>9X=X-3
350 RETURN
360 IF A+B>9X=X-3;CX=X;RETURN
370 IF AbB>9X=X-3;IF AbB>99X=X-3
380 CX=X;RETURN
390 CX=-18;PRINT "CORRECT!";R=R+1;FOR D=1TO 200;NEXT D;GOTO 110
400 CY=-32;PRINT " CORRECT ANSWER IS ";RETURN
410 FOR D=1TO 500;NEXT D;W=W+1;RETURN
420 FOR D=0TO 176;&(10)=D;NEXT D;RETURN
430 FOR D=176TO 0STEP -1;&(10)=D;NEXT D

```

```
440 BC=RND (32)*8;FC=BC-1;RETURN
450 CX=-40;RETURN
460 PRINT ;PRINT "*>      SESSION OVER      <*>;PRINT ;PRINT " RIGHT ANSWERS: ",#
1,R;PRINT " WRONG ANSWERS: ",#1,W
470 PRINT ;PRINT ;PRINT " FOR A REPEAT, PUSH....1";PRINT " FOR DIFFERENT, PUSH
....2";GOSUB E
480 FOR D=1TO 2000;IF &(23)=8GOTO 100
490 IF &(22)=8GOSUB F;GOTO 40
500 NEXT D;FC=BC-7;GOTO 480
```


1
2
3
4
6
7
8
9

```
0 CLEAR : NT=0 : CY=0 : PRINT " HOW MANY PLAYERS? (1-3)
```

```
30 CY=27;CX=-62;PRINT "SCORE:";BOX 0,22,160,1,1;CY=11;CX=-48;PRINT "TRY LESS THAN"
```

```
50 M=0;V=0;L=100;R=RND (99)
```

```
70 IF X=49GOTO 110
```

```
100 GOSUB 480;GOSUB 490;INPUT "C"
```

```

120 IF R=A D=D+1;GOSUB 460;GOSUB 510
130 IF R=B E=E+1;GOSUB 470;GOSUB 510

```

```
130 IF R=B: E=E+1; GOSUB 470; GOSUB 510
140 IF R=C: F=F+1; GOSUB 480; GOSUB 510
150 IF R=D: G=G+1; GOSUB 490; GOSUB 510
```

```
150 IF V>0FOR Z=1TO 1000;NEXT Z
```

```
160 GOSUB 460;GOSUB 500;GOSUB 470;GOSUB 500;GOSUB 480;GOSUB 500
```

170 G=A>B ; H=A<B ; I=A>C ; J=A<C ; K=A>R ; N=A<R ; O=B>C ; P=B<C ; Q=B>R ; S=B<R ; U=C>R ; W=C<R

```
180 IF X=50GOTO 230
```

```
190 IF X=51GOTO 300
```

200 IF K L=A

210 IF N M=A

```
220 GOTO 380
```

230 IF K IF Q IF G L=R

240 IF K IF Q IF H L=A

```
250 IF K IF S L=A ;M=B
```

```
260 IF N IF Q L=B ; M=A
```

270 IF N IF S IF G M=A

280 IF N IF S IF H M=B

```
290 GOTO 380
```

```
300 IF K IF Q IF U GOSUB 530
```

```
310 IF K IF Q IF W GOSUB 600
```

```
320 IF K IF S IF U GOSUB 640
```

```
330 IF N IF Q IF U GOSUB 680
```

```
340 IF N IF S IF W GOSUB 720
```

```
350 IF N IF S IF U GOSUB 790
```

```
360 IF N IF Q IF W GOSUB 830
```

```
370 IF K IF S IF W GOSUB 870
```

```
380 IF D>9GOSUB 460;GOSUB 520
```

```
390 IF E>9GOSUB 470;GOSUB 520
```

```

000 IF F>9GOSUB 48
010 IF T>0GOTO 440

```

```
10 IF T>0GOTO 44
```

20 IF $V > 0$ G

30 GOTO 60

```
40 FOR Z=1
```

50 GOTO 10

```
460 CY=-12;CX=-38;RETURN
470 CY=-21;CX=-38;RETURN
480 CY=-29;CX=-38;RETURN
490 PRINT " YOUR TURN!";RETURN
500 PRINT " ";RETURN
510 PRINT "MATCHED MY NUMBER!";U=U+1;RETURN
520 PRINT "IS A WINNER!!!!";T=T+1;RETURN
530 IF H IF J L=A
540 IF G IF P L=B
550 IF I IF O L=C
560 IF H IF B=C L=A
570 IF P IF A=C L=B
580 IF I IF A=B L=C
590 RETURN
600 IF H L=A ;M=C
610 IF G L=B ;M=C
620 IF A=B L=A ;M=C
630 RETURN
640 IF J L=A ;M=B
650 IF I L=C ;M=B
660 IF A=C L=A ;M=B
670 RETURN
680 IF P L=B ;M=A
690 IF O L=C ;M=A
700 IF B=C L=B ;M=A
710 RETURN
720 IF G IF I M=A
730 IF H IF O M=B
740 IF J IF P M=C
750 IF G IF B=C M=A
760 IF O IF A=C M=B
770 IF J IF A=B M=C
780 RETURN
790 IF G M=A ;L=C
800 IF H M=B ;L=C
810 IF A=C M=A ;L=C
820 RETURN
830 IF I M=A ;L=B
840 IF J M=C ;L=B
850 IF A=C M=A ;L=B
860 RETURN
870 IF O M=B ;L=A
880 IF P M=C ;L=A
890 IF B=C M=B ;L=A
900 RETURN
```

2333 W. Huron
Chicago, IL 60612
February 11, 1982
(312) 489-1832

Ji O
Marselle

Robert Fabris
3626 Morris Drive

San Jose, CA 95127-9990

works @
IMAGE?

Dear Robert,

I finally was able to make a tape of the programs which I wrote to do line drawing and moving, scaling, and rotation. If you recall, I sent you a listing which was produced on a different computer, since I didn't have access to a dally printer. I think the xerox I enclose is probably as readable.

Let me recap what all the programs (actually subprograms; they all RETURN) do. The idea is that we store (X, Y, P) coordinates, i.e. LINE X, Y, P , ~~into~~ by packing them into one integer @array element.

This packing is done by PACK (entry line 900) and unpacked by UNPK (500). @ (0) contains the number of points to follow in @ (1), @ (2), ..., @ (@ (0)).

The array can be loaded from an algorithm or by using LORAN (1500), which is a line drawing program.

$JX(1)$ and $JY(1)$ control the (x, y) coordinates of the rubber band line. $TR(1)$ is used to save the current (x, y) coordinates. If $KN(1) < \phi$ when $TR(1)$ is pulled, $p = \phi$. Otherwise $p = 1$. Press a key in column 1 of the keypad to delete the last drawn point, a key in column 2 to exit.

The picture may then be transformed by translation in the x & y direction, scaling up (can be modified to scale down but can't do both together) and various rotations. If rotation is to be done, sines & cosines are required. $COSSLO(1000)$ was written by Dan Sandin when I took a course with him and uses $@(0) \rightarrow @(90)$ to hold the cosines of $0^\circ * 250$ to $90^\circ * 250$ (~~the~~ $* 250$ is for accuracy since no fractions in BR). Thus the vectors will need to use $@(91)$, ... and $LDRAW$ will need to be changed (in fact that's the way it is now). $COSSIN(700)$ calculates the sines and cosines of its arg (any value in degrees, its normalized to $0^\circ \rightarrow 360^\circ$).

Needless to say if any program is modified, the
@() array needs to be reloaded. ②

SETMOV(2500) interacts with the user to move a
picture and calls MOV(2000) to do the transformation.

SETSCALE(2200) does scaling and calls SCALE(2100).

ROT(2300) does rotation and calls XROT(1200),

YROT(1300), ZROT(1400), or XYROT(2400) to perform
rotations. Other combinations are possible (XZ, YZ, XYZ, XZY...).

The transformations are obtained from Newman &
Sproull's Principles of Interactive Computer Graphics,
2nd edition.

Now for the tape layout.

SIDE 1: Lines 497 (leader) - 2050 2 copies

SIDE 2: Lines 2097 - 2590 2 copies

Following are clean (almost no comments
or PRINT statements) versions of UNPK, PACK, LDRAW,
SCALE, SETSCALE, MOV, SETMOV, COSSIN, COSLD,
XROT, YROT, ZROT, ROT and a
truncated XYROT (tape ran out).

Maybe some folks might make use of
these programs and in the process learn about
elementary computer graphics transformations.

Have fun!

James P. Marrella

2333 W. Huron

Chicago, IL 60612

October 23, 1980

Robert Fabris
3626 Morrie Drive
San Jose, CA 95127.

Dear Bob,

I've written a bunch of BASIC programs to allow users to interactively construct line drawings with the joystick and to rotate, scale, and translate them once they've been drawn. The documentation is different in that I typed copies of all my programs on a minicomputer. This was extra work since any change in the BASIC code had to be updated on the mini, but it was the only way to get printouts easily.

The printout I've sent contains all the programs in line number order and the comments should be adequate to explain what's going on. Basically LDRAW (line 1500) lets you use joystick 1 to draw lines with mode 0 or 1. Each xy coordinate and drawing mode is packed into the @ array by PACK.

To scale a picture up (scaling down is a little trickier and you can't do both in the same program because of non-linearities) you use SETSCALE (2200), which reads the joystick and calls UNPK to unpack each stored xy coordinate. Then we call SCALE to perform the transformation.

Obviously all the programs don't need to be in memory at once and in fact they probably won't all fit. There's lots of BASIC comment lines that can be removed. Of course any change requires you to use LDRAW to refill the @ array, since its location changes.

Since rotation requires using sines and cosines, I store the cosines of $0 \rightarrow 90$ degrees scaled up by 250 (cosines are < 1 and BASIC doesn't handle fractions) using a program Dan Sandin got from someone (COSLD). So it's necessary to GOSUB 1000 to set up the cosine array whenever the BASIC code is modified. So the packed xy coordinates from LDRAW start at 0(91). If you're not using the rotate code, the code can be changed to use all of the array elements for packed xy coordinates and COSLD and COSSIN can be removed.

The COSSIN program calculates the sine and cosine of any angle (positive or negative). We only need to store the 91 cosines since all others can be derived from these. If you're doing rotates, the variable R is used in a computed GOSUB (line 2385), so for example if you want to rotate about the x axis, set $R=1200$ (entry point to XROT). I've also got YROT (1300), ZROT (1400), and I derived the equations for x followed by y rotation in XYROT (2400). Note that y followed by x is not the same. Ambitious folks who enjoy multiplying matrices can derive other equations for xz, yz, xyz, xzy, etc. rotations.

In each rotation I give the formula for the new x and y coordinate (x' and y') in terms of the old x, y, and z. Since for now LDRAW works in 2D, z is always 0. In line 2410 and 2420 what I'm giving are the equations for the new x and y when we rotate about the x axis by the angle t (theta) and about the y axis by the angle g (gamma).

P.S. Here's the \$12.50 for my renewal.
Thanks.

Jim Marselle

UNPK

Jim Marselle 6/27/80

This program unpacks x, y, and p from the value in @i).

needs:

@i).

returns:

x, y, p.

clobbers:

a.

calls:

nothing.

500 .unpk x, y, & p.

510 a = @i)

520 p = 1

530 if a < 0 p = 0 ; a = -a

540 x = a / 100 - 80 ; y = rm - 50

550 return

COSSIN

Jim Marselle 5/27/80

This program calculates the sin and cos of t in c and s, resp.

needs:

t.

returns:

c, s.

clobbers:

u, v.

calls:

nothing.

700 .cos & sin of t in c

710 .& s. clobbers u & v.

720 u = t / 360 ; u = rm

730 v = abs(u)

740 .cos(-t) = cos(t)

750 if v < 91 c = @v) ; goto 800

760 if v < 181 c = -@v - 180) ; goto 800

770 if v < 271 c = -@v - 180) ; goto 800

780 c = @v - 360 - v)

790 .sin(t) = cos(90 - t)

791 .sin(-t) = -cos(90 - t)

800 if v < 91 s = @v - 90) ; goto 850

810 if v < 181 s = @v - 90) ; goto 850

820 if v < 271 s = -@v - 270 - v) ; goto 850

830 s = -@v - 270)

840 if u < 0 s = -s

860 return

PACK

Jim Marselle 5/27/80

This program packs x, y, and p into @i).

needs:

i, p, x, y.

returns:

@(i) = packed value.

clobbers:

nothing.

calls:

nothing.

900 .pack p, x, y into @(i)

910 .100 * (x + 80) + y + 50 <

920 if p = 1 @(i) = 100 * x + y + 8050 ; return

930 @(i) = -100 * x - y - 8050 ; return

COSLP

Dan Sandin

5/27/80

This program loads @(0) -> @(90) with the cosines of 0 -> 90 degrees times 250 (to keep roundoff as small as possible).

needs:

enough @() space.

returns:

nothing.

clobbers:

m, q, v, x.

calls:

nothing.

1000 .cos array load

1010 x = 10000

1020 m = 2715

1030 v = 0

1060 @(0) = 250

1070 for q = 1 to 90

1080 v = v - x / m

1090 x = x + v

1095 @(q) = x / 40 ; * 250

1100 next q

1110 return

XROT

Jim Marselle

5/27/80

This program calculates transformed coordinates for a rotation about the x axis at the angle whose cosine = c and sine = s.

needs:

x, y, z = original pix coordinates. (note: z = 0 if we're 2D).
c and s.

returns:

u, v = new pix x-y coordinates.

clobbers:

nothing.

calls:

```
nothing.
*****
1200 .x rot, new x, y, z in
1210 .u, v, w, needs c & s
1220 .x' = x, y' = ycos + zsin, z' = zcos - ysin
1230 u = x
1240 v = (y * c + z * s) / 250
1250 return
*****
```

YROT

Jim Marselle 5/27/80

This program calculates transformed coordinates for a rotation about the y axis at the angle whose cosine = c and sine = s.

needs:

x, y, z = original pix coordinates. (note: z = 0 if we're 2D).
c and s.

returns:

u, v = new pix x-y coordinates.

clobbers:

nothing.

calls:

nothing.

```
*****
1300 .y rot
1310 .x' = xcos - zsin, y' = y, z' = xsin + zcos
1320 u = (x * c - z * s) / 250
1330 v = y
1340 return
*****
```

ZROT

Jim Marselle 5/27/80

This program calculates transformed coordinates for a rotation about the z axis at the angle whose cosine = c and sine = s.

needs:

x, y, z = original pix coordinates. (note: z = 0 if we're 2D).
c and s.

returns:

u, v = new pix x-y coordinates.

clobbers:

nothing.

calls:

nothing.

```
*****
1400 .z rot
1410 .x' = xcos + ysin, y' = ycos - xsin, z' = z
1420 u = (x * c + y * s) / 250
1430 v = (y * c - x * s) / 250
1440 return
*****
```

LDRAW

Jim Marselle 6/2/80

This pgm allows the user to draw lines with mode 0 (no draw) and mode 1 (draw). Use jx(1) and jy(1) to control the rubber-band line. pull tr(1) to draw the line, if kn(1) < 0, mode = 0. Otherwise mode = 1.
To delete the last drawn point hit a key in column 1, to exit hit a key in column 2.

needs:

Enough @() space to store the packed x, y, and p values starting at @ (92) (the number of points will be stored at @ (91)). This is done to allow @ (0) -> @ (90) to contain the cosines of 0 -> 90 degrees for rotations. If rotation will not be used, change this pgm.

returns:

@() filled with the packed values of x, y, and p (the draw mode).
b and c are the x and y center of the pix, resp.

clobbers:

b, c, d, e, i, p, u, v, x, y.

calls:

PACK, UNPK.

```

1500 .line draw
1510 nt = 1
1560 clear
1570 cy = 40
1580 print "jx(1) , jy(1) to move"
1590 print "tr(1) to draw line"
1600 print "if kn(1) > 0 , we draw(p = 1)"
1610 print "else we don't.(p = 0)"
1620 print "hit 1 to delp" delete point
1630 print "2 to quit"
1640 i = 92 ; x = 0 ; y = 0 ; p = 0
1641 .xmax, xmin, ymax, ymin
1642 b = -80 ; c = 79 ; d = b ; e = c
1650 .pack dummy first point into @ (92)
1660 gosub 900
1670 line 0 , 0 , 0
1680 input b ; clear
1685 u = 0 ; v = 0 ; i = 93
1690 .quit?
1700 if &(22) @ (91) = i - 92 ; goto 1890
1710 u = u + jx(1) ; v = v + jy(1)
1720 line u , v , 1 ; line x , y , 0
1721 line u , v , 2 ; line x , y , 0
1740 if tr(1) goto 1820 ; .line
1750 if &(23) = 0 goto 1700
1760 .delp
1770 if i = 93 goto 1700
1780 i = i - 2
1790 gosub 500 ; .unpk x , y , p
1791 line x , y , 0
1792 u = x ; v = y
1793 i = i + 1
1795 gosub 500
1800 if p line x , y , 2 ; line u , v , 0
1805 x = u ; y = v
1810 if &(23) mu = 100 ; goto 1810
1812 goto 1700
1820 if kn(1) > 0 p = 1 ; goto 1840
1830 p = 0
1840 x = u ; y = v
1850 line x , y , p
1860 gosub 900
1861 if x > b t = x ; goto 1863

```

```

1862 if x < c c = x
1863 if y > d d = y ; goto 1870
1864 if y < e e = y
1870 i = i + 1
1871 if tr(1) mu = 100 ; goto 1871
1880 goto 1700
1890 clear
1891 .get xcen & ycen
1892 b = (b + c) / 2
1893 c = (d + e) / 2
1900 for i = 92 to @ (91) + 91
1910 gosub 500
1920 line x , y , p
1930 next i
1940 return

```


MOV

Jim Marselle 6/2/80

This program calculates the transformed x-y coordinates for a translation of k units in x, l units in y, and returns the transformed coordinates in u and v, resp.

needs:

k, l, x, and y.

returns:

u and v = new x-y coordinates.

clobbers:

nothing.

calls:

nothing.

```

2000 .mov
2010 .x' = x + tx
2020 .y' = y + ty
2030 u = x + k
2040 v = y + l
2050 return

```

SCALE

Jim Marselle 6/2/80

This program calculates the transformed x-y coordinates for a scale (up) of factor k in x and l in y. We return the transformed coordinates in u and v, resp. Since we assume the pix is not centered at the origin we actually perform a concatenation of the transformations translate-to-origin, scale, and translate-back. To do this we need the x center and y center of the pix in b and c, resp.

needs:

b, c, k, and l.

returns:

u and v, the new x-y coordinates.

clobbers:

nothing.

calls:

nothing.

```

2100 .scale
2110 .x' = xSx + Tx(Sx - 1)
2120 .y' = ySy + Ty(Sy - 1)
2130 u = (x - b) * k + b
2140 v = (y - c) * l + c
2150 return
*****
*****

```

SETSCALE Jim Marselle 6/2/80

This program allows the user to scale up a pix in the x and y dimensions by using kn(1) (We could use kn(2) for y scaling). When the user pulls tr(1) another copy of the pix is drawn. Hit a key in column one to exit.

needs: @() set up.

returns: nothing.

clobbers: i, k, l, p, u, v, x, y.

calls: UNPK, SCALE.

```

2200 .set scale
2205 if S(23) return
2210 if tr(1) = 0 goto 2205
2215 if tr(1) mu = 100 ; goto 2215
2220 . k = kn(1) + 129 ; .down
2230 k = kn(1) / 40 + 4 ; .up
2235 l = k
2240 for i = 92 to @ (91) + 91
2245 gosub 500
2250 gosub 2100
2255 line u , v , p
2260 next i
2270 goto 2205
*****

```

ROT

Jim Marselle 6/2/80

This program rotates a pix about the axis specified by r (r is the entry point of the desired coordinate transformation routine, e.g. XROT, YROT, etc.) We use kn(1), kn(2), and kn(3) to specify the rotation angle about the x, y, and z axes, resp. These angles are saved in t (theta), g (gamma), and p (phi). The sines of t, g, and p are s, f, and h, resp. The cosines of t, g, and p are c, d, and e, resp. To draw a rotated pix, pull tr(1), to exit hit any key in column 1.

needs: @() set up and r, the entry point of the desired rotation routine.

returns: nothing.

clobbers: c, d, e, f, h, i, p, t.

calls: COSSIN, UNPK, and the ROT routine specified by r.

```

2300 .rot
2310 if s(23) return
2320 if tr(1) = 0 goto 2310
2330 if tr(1) mu = 100 ; goto 2330
2340 t = kn(1) * 180 / 128
2341 g = kn(2) * 180 / 128
2342 p = kn(3) * 180 / 128
2343 a = t ; t = g ; gosub 700
2344 d = c ; f = s
2345 t = p ; gosub 700
2346 e = c ; h = s
2347 t = a ; gosub 700
2360 for i = 92 to G(91) + 91
2380 gosub 500
2385 gosub r
2390 line u , v , p
2395 next i
2396 goto 2310

```


XYROT

Jim Marselle 6/2/80

This program calculates transformed coordinates for a rotation about the x followed by y axes at the angles whose cosines = c and d, resp. and whose sines = s and f, resp.

needs:

x, y, z = original pix coordinates. (note: z = 0 if we're 2D).
c, s, d, f.

returns:

u, v = new pix x-y coordinates.

clobbers:

nothing.

calls:

nothing.

```

2400 .xyrot
2410 .x' = xcosg + ysint sing - zsing cost
2420 .y' = ycost + zsint
2430 . cos t, g, p = c, d, e. sin t, g, p = s, f, h.
2440 u = x * d / 250
2441 u = u + (y * s / 250 * f / 250)
2442 u = u - (z * f / 250 * c / 250)
2450 v = y * c / 250 + z * s / 250
2460 return

```


SETMOV

Jim Marselle 6/2/80

This program allows the user to translate a pix in the x and y directions by using jx(1) and jy(1), resp. We draw a flashing box which indicates the center of where the pix will be drawn. When the user pulls tr(1) another copy of the pix is drawn. Hit a key in column one to exit.

needs:

@() set up, b and c = x center and y center, resp.

returns:

nothing.

clobbers:

i, k, l, m, n, p, u, v, x, y.

calls:

UNPK, MOV.

```
2500 .setmov
2505 m = 0 ; n = 0
2510 if &(23) return
2515 m = m + jx(1) ; n = n + jy(1)
2520 box m , n , 2 , 2 , 1
2525 box m , n , 2 , 2 , 3
2530 if tr(1) = 0 goto 2510
2535 if tr(1) mu = 100 ; goto 2535
2540 k = m - b ; l = n - c
2545 for i = 92 to @(91) + 91
2550 gosub 500
2560 gosub 2000
2570 line u , v , p
2580 next i
2590 goto 2510
```
